



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/626,273	07/24/2003	Naga A. Ayachitula	SVL920030043US1	9607

7590 08/22/2006  
Michael E. Hudzinski  
FAY, SHARPE, FAGAN, MINNICH & McKEE, LLP  
Seventh Floor  
1100 Superior Avenue  
Cleveland, OH 44110-2518

EXAMINER
----------

VAUTROT, DENNIS L

ART UNIT	PAPER NUMBER
----------	--------------

2167

DATE MAILED: 08/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b> 10/626,273	<b>Applicant(s)</b> AYACHITULA ET AL.	
	<b>Examiner</b> Dennis L. Vautrot	<b>Art Unit</b> 2167	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

#### Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

#### Status

- 1) ☒ Responsive to communication(s) filed on 26 May 2006.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

#### Disposition of Claims

- 4) ☒ Claim(s) 1-34 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-34 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

#### Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 24 July 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

#### Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

#### Attachment(s)

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)             | 4) <input type="checkbox"/> Interview Summary (PTO-413)                     |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)    | Paper No(s)/Mail Date. _____  |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____   | 6) <input type="checkbox"/> Other: _____                                    |

## **DETAILED ACTION**

### ***Response to Amendment***

1. The applicants' amendment, filed 26 May 2006, has been received, entered into the record and considered.
2. As a result of the amendment, claims 1, 9, 21, and 26 – 27 are amended. Claims 31 – 34 have been added. Claims 1 – 34 are pending in the application.

### ***Response to Arguments***

3. Applicant's arguments with respect to claims 1 – 20 have been considered but are moot in view of the new ground(s) of rejection.
4. Applicant's arguments with respect to claims 21 – 26 have been considered but are moot in view of the new ground(s) of rejection.
5. Applicant's arguments with respect to claims 27 – 30 have been considered but are moot in view of the new ground(s) of rejection.

### ***Claim Rejections - 35 USC § 102***

6. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

7. Claims 1, 7, 21, 24 – 28, and 30-31 are rejected under 35 U.S.C. 102(e) as being anticipated by **Rappold, III** (hereinafter, **Rappold**, US 2004/0117397).

8. Regarding claim 1, **Rappold** discloses an abstraction layer for a database containing database records each including a plurality of fields [attributes] stored in one or more tables, the fields being associated with each associated record [logical entity] by a key [entity ID] disposed in at least one key column of each of the one or more tables (See page 2, paragraph [0024] "To link those logical entities to their attributes, each logical entity is given an entity ID (ID#)."), the abstraction layer comprising:

a key column identifier [FIG. 1 pointer to 110 and 112] that identifies the at least one key column [columns 110 and 112] (See page 2, paragraph [0028] "Column 110 holds numbers identifying the individual row 100, similar to a typical relational database... Column 112 holds entity Ids identifying the individual who has an attribute stored on that row." This shows two different key columns, one for the entities and one for each attributes [or fields].); and

one or more metadata tables [11] containing metadata relating to the database (See page 2, paragraph [0029] "The other component of the extensible database

system shown in FIG. 1 is metadata table 11.”), the one or more metadata tables including at least:

a controls table [10] containing control records corresponding to fields [100] of the database (See page 2, paragraph [0028] “Column 110 holds numbers identifying the individual row 100, similar to a typical relational database.” Each of these individual rows correspond to fields of the database.), the control record for each field including at least a control key [110] associating the control record with the field and at least one metadatum [114] corresponding to the field (See page 2, paragraph [0028]. Column 110 holds the control key associating the record with the field. Also, “Column 114 holds attribute IDs representing the attribute type stored on that row.” This represents the metadatum corresponding to the field. Column 114 points to the section of the metadata table which holds the attribute names and attribute types.);

wherein the abstraction layer provides a database interface substantially independent of an underlying model of the database thereby providing extensible functionality for the database (See page 1, paragraph [0008] “Another technical advantage of particular embodiments of the present invention is that the extensible database system and method enables a user to store new data in a table without having to create new columns or define a new table using DDL. As a result, the behavior of database applications can be adjusted during run-time, instead of having to restrict the persistence layer processing only to what can be handled by the existing schema.” This clearly shows substantial independence from the schema of the database.)

9. Regarding claim 7, **Rappold** additionally teaches the one or more tables includes at least two tables (See FIG. 1 showing a Metadata table and FIG. 2, showing a data table), and the control record for each field [200a] further includes a table name that in combination with the control key [210] associates the control record with the field (FIG. 2, item 200a represents a control record for a field, with column 210 being the control key and column 214 representing the table name.)

10. Regarding claim 21, **Rappold** discloses a method for accessing a database containing database records each including a plurality of fields stored in one or more tables, the method comprising: formulating a database access command [SQL statements] using metadata [different parameter values] related to the database contained in an abstraction layer (See page 4, paragraph [0049] "Because a standard format is used, the same query can be used repeatedly with different parameter values, which results in faster response times for batch-prepared SQL statements that are parsed one time and cached by the database server engine." The different parameter values are found in the metadata contained in the abstraction layer.),

the metadata for each database field being accessible using an abstraction layer control record [200a] associated with the database field [216] (See page 4, paragraph [0050] "Transformation logic can also be provided to convert data from a conventional relational data format to the extensible data format using metadata and unique algorithms." Here, the metadata in the control record, located in FIG. 2, item 200a, for example is transformed by accessing the metadata in the control record.);

and executing the formulated database access command [process a query result] to access the database (See page 4, paragraph [0050] "Transformation module 62 can query the extensible database, comprising data table 60 and metadata table 61, process a query result, and transform the result into objects for processing."),

the abstraction layer providing a database interface substantially independent of an underlying model of the database (See page 1, paragraph [0008] "Another technical advantage of particular embodiments of the present invention is that the extensible database system and method enables a user to store new data in a table without having to create new columns or define a new table using DDL. As a result, the behavior of database applications can be adjusted during run-time, instead of having to restrict the persistence layer processing only to what can be handled by the existing schema." This clearly shows substantial independence from the schema of the database.)

11. Regarding claim 24, **Rappold** additionally discloses the abstraction layer includes: a controls table [10] containing the control records [100] of the database fields [110] (See page 2, paragraph [0028] "Column 110 holds numbers identifying the individual row 100, similar to a typical relational database." Each of these individual rows correspond to fields of the database.), each control record including a field key [110, 112] (See page 2, paragraph [0028] "Column 110 holds numbers identifying the individual row 100, similar to a typical relational database... Column 112 holds entity Ids identifying the individual who has an attribute stored on that row." This shows two different key columns, one for the entities and one for each attributes [or fields].); and at

least one metadata table [20] containing records corresponding to database fields and linked to the control record by the field key [210] (See FIG 2. showing the metadata table containing records corresponding to database fields, with the field key being listed in column 210, which links 110 of FIG. 1.)

12. Regarding claim 25, **Rappold** additionally discloses a controls table [10] containing the control records [100] of the database fields, each control record including at least one index metadatum [attribute ids] (See page 2, paragraph [0028]. Column 110 holds the control key associating the record with the field. Also, "Column 114 holds attribute IDs representing the attribute type stored on that row." This represents the metadatum corresponding to the field. Column 114 points to the section of the metadata table which holds the attribute names and attribute types.); and at least one additional metadata table [11] containing indexed metadata associable with database fields by the at least one index metadatum [114] of the control records (See the second chart of FIG 1, (11), where the index column made up of elements 111a, and 111c are located on the left being associated with the fields from the other table in column 114.)

13. Regarding claim 26, **Rappold** additionally discloses executing a user application program, the formulating of a database access command [database requests] being performed as an operation of the executing user application program [application]. (See page 5, paragraph [0053] "ActionBean 701 encapsulates information that is used by an application for processing database requests." And see paragraph [0055] "The

TransformActionToMeta algorithm extracts the value of the ActBean from ActionBean 701. It then retrieves the metadata for ActionBean 701 by matching the value of ActBean to the entity name value in the retrieved metadata information, and saves the metadata information in MetadataBean.” TransformActionToMeta is part of the application mentioned in the earlier line.)

14. Regarding claim 27, **Rappold** additionally discloses an article of manufacture comprising one or more program storage media readable by a computer and embodying at least an abstraction layer for facilitating accessing a database containing database records each including a plurality of fields stored in one or more tables (See generally FIG 1.),

the abstraction layer including: a control table [10] containing control records corresponding to database fields [110], each control record containing metadata [114] associated with the corresponding database field [110] (See FIG 1. showing the control table with control records. The column 114 holds a number that represents the metadata associated with the corresponding database field. Particularly, that number is a reference to a separate table with even further metadata corresponding to the field. Alternatively, just looking at the second table of FIG. 1, that alone could be considered a control table containing records corresponding to database fields as well as the metadata – the name and type are shown in column 113 and 115 respectively.),

and at least one additional table containing additional metadata [11], each database field being selectively associated with one or more selected portions of the

additional metadata through metadata contained in the control record corresponding to the database field. (See the second chart of FIG 1, (11) where the index column made up of elements 111a, and 111c are located on the left being associated with the fields from the other table in column 114.)

wherein the abstraction layer provides a database interface substantially independent of an underlying model of the database (See page 1, paragraph [0008] "Another technical advantage of particular embodiments of the present invention is that the extensible database system and method enables a user to store new data in a table without having to create new columns or define a new table using DDL. As a result, the behavior of database applications can be adjusted during run-time, instead of having to restrict the persistence layer processing only to what can be handled by the existing schema." This clearly shows substantial independence from the schema of the database.)

15. Regarding claim 28, **Rappold** additionally discloses formulating a database access command; and during the formulating, accessing an abstraction layer to identify at least one constraint [data in the table] on the database access command (See page 4, paragraph [0050] "FIG. 6 illustrates and extensible database system in which transformation module 62 is operable to import and export data into and out of data table 60 and metadata table 61" [formulating the database access command.] And "Transformation module 62 can query the extensible database, comprising data table 60 and metadata table 61, process a query result, and transform the result into objects for

processing.” This is accessing the abstraction layer with the constraint being based on the data.)

16. Regarding claim 30, **Rappold** additionally discloses the article of manufacture further embodies a user application program [application] executable by the computer, the executing user application program being operatively linked with the method [ActionBean] for accessing the database. (See page 5, paragraph [0053] “ActionBean 701 encapsulates information that is used by an application for processing database requests.” And see paragraph [0055] “The TransformActionToMeta algorithm extracts the value of the ActBean from ActionBean 701. It then retrieves the metadata for ActionBean 701 by matching the value of ActBean to the entity name value in the retrieved metadata information, and saves the metadata information in MetadataBean.” TransformActionToMeta is part of the application mentioned in the earlier line.)

17. Regarding claim 31, **Rappold** additionally discloses the abstraction layer as set forth in claim 1, wherein said extensible functionality for the database provided by the abstraction layer includes:

one or more modifiable metadata tables containing metadata relating to the database (See FIG. 1 showing the metadata tables),

the one or more modifiable metadata tables including at least: a modifiable controls table [10] containing control records [100] corresponding to fields of the database (See page 2, paragraph [0028] “Column 110 holds numbers identifying the

individual row 100, similar to a typical relational database.” Each of these individual rows correspond to fields of the database.);

the control record for each field including at least a control key [110] associating the control record with the field and at least one metadatum [114] corresponding to the field (See page 2, paragraph [0028]. Column 110 holds the control key associating the record with the field. Also, “Column 114 holds attribute IDs representing the attribute type stored on that row.” This represents the metadatum corresponding to the field. Column 114 points to the section of the metadata table which holds the attribute names and attribute types.);

wherein the modifiable controls table can be modified to provide different or additional metadata for selectively extending the functionality of said database interface (See page 1, paragraph [0008] “Another technical advantage of particular embodiments of the present invention is that the extensible database system and method enables a user to store new data in a table without having to create new columns or define a new table using DDL. As a result, the behavior of database applications can be adjusted during run-time, instead of having to restrict the persistence layer processing only to what can be handled by the existing schema.” This shows the modifiable table is extending the functionality of the database.)

### ***Claim Rejections - 35 USC § 103***

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

Art Unit: 2167

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

19. Claims 2 and 3 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 1 above, and further in view of **Helgeson et al.** (US 6,643,652).

20. Regarding claim 2, **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the at least one metadatum of at least one control record includes a datatype index value indicative of a datatype of the corresponding field, and the one or more metadata tables further include: a datatypes table associating a plurality of datatype indices with datatypes. However, **Helgeson et al.** teaches the at least one metadatum of at least one control record includes a datatype index value indicative of a datatype of the corresponding field, and the one or more metadata tables further include: a datatypes table associating a plurality of datatype indices with datatypes. (See column 16, lines 23-25 "Attr\_type Int Y The number corresponds to the data type of the attribute..." and column 17, lines 15-35 – see generally the chart where different datatypes are defined.) It would have been obvious to one with ordinary skill in the art to use an index value for the datatypes in order to allow new datatypes to be added at any time by just adding another field in the table. It is for this reason that one of ordinary skill in the art would have been motivated to have the at least one metadatum of at least one control record include a datatype index value indicative of a datatype of the corresponding field, and the one or more

metadata tables further include: a datatypes table associating a plurality of datatype indices with datatypes.

21. Regarding claim 3, **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the datatypes of the datatypes table are selected from a group including: a character datatype, a numeric datatype, a text data type, a date data type, a time datatype, and a timestamp datatype. However, **Helgeson et al.** teaches the datatypes of the datatypes table are selected from a group including: a character datatype, a numeric datatype, a text data type, a date data type, a time datatype, and a timestamp datatype. (See column 22, lines 41-68, where datatypes for the object are assigned to the various fields). It would have been obvious to one with ordinary skill in the art to include these various datatypes in the abstraction layer as they are some of the most commonly used. It is for this reason that one of ordinary skill in the art would have been motivated to have the datatypes of the datatypes table selected from a group including: a character datatype, a numeric datatype, a text data type, a date data type, a time datatype, and a timestamp datatype.

22. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** in view of **Helgeson et al.** (US 6,643,652) as applied to claim 2 above, and further in view of **Subramaniam et al.** (US 6,782,383). **Rappold** and **Helgeson et al.** teach an abstraction layer for a database substantially as claimed. **Rappold** and **Helgeson et al.** fail to teach an operators table associating a database operation with a database type

index value and with a corresponding display operator. However, **Subramaniam et al.** teaches an operators table [picklist] associating a database operation with a database type index value and with a corresponding display operator. (See column 21 lines 62-67 "In one embodiment, Search Term Separator control includes a picklist with the following options: "All words (AND match)", "Any word (OR match)," and "Exact Phrase". "All words (AND match)" option specifies a search to find records or documents containing every search keyword entered by the user." Here the database operators are "and", "or", and "Exact Phrase".) It would have been obvious to one with ordinary skill in the art that by including a table with the database operations, there could be different output based on the particular data type. It is for this reason that one of ordinary skill in the art would have been motivated to include an operators table associating a database operation with a database type index value and with a corresponding display operator.

23. Claim 5 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** in view of **Helgeson et al.** (US 6,643,652) and in view of **Subramaniam et al.** (US 6,782,383) as applied to claim 4 above, and further in view of **Williams et al.** (US 6,934,696). **Rappold, Helgeson et al.,** and **Subramaniam et al.** teach an abstraction layer for a database substantially as claimed. **Rappold, Helgeson et al.,** and **Subramaniam et al.** fail to teach the operators table further associates the database operation with a corresponding second display operator, and the operators table further includes: a language field associating a different language with each of the

Art Unit: 2167

corresponding display operator and the corresponding second display operator, whereby the database operation has associated therewith display operators in at least two different languages. However, **Williams et al.** teaches the operators table further associates the database operation with a corresponding second display operator, and the operators table further includes: a language field [text identifier] associating a different language with each of the corresponding display operator and the corresponding second display operator, whereby the database operation has associated therewith display operators in at least two different languages [in any language]. (See column 26, lines 64-67 "The reason the system uses a text identifier is that it can define text in the database as a number while having the actual text stored in the string table in a language module. This is done for internationalization." And see column 27, lines 38-42 "...stored as text identification numbers which can later be easily translated by code that associates the text characters to stored text identification numbers for obtaining a database that can be written in any language.") One with ordinary skill in the art would have combined the display operator table as mentioned earlier with an associated language field table in order to allow for different language versions with minimal overhead. Here, only a number or code would need to be passed, representing a string of text, rather than the actual string. The string that is displayed will depend on what language is set. It is for this reason that one of ordinary skill in the art would have been motivated to have the operators table further associate the database operation with a corresponding second display operator, and the operators table further includes: a language field associating a different language with

each of the corresponding display operator and the corresponding second display operator, whereby the database operation has associated therewith display operators in at least two different languages.

24. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** in view of **Helgeson et al.** as applied to claim 2 above, and further in view of **Williams et al.** (US 6,934,696). **Rappold** and **Helgeson et al.** teach an abstraction layer for a database substantially as claimed. **Rappold** and **Helgeson et al.** fail to teach the datatype index value indicates that the corresponding field is numeric, and the at least one metadatum further includes: a sub-datatype index value indicative of a type of numeric value of the corresponding field, the sub-datatype index value being selected from a group including at least integer and floating-point numeric value types. **Williams et al.**, however, teaches the datatype index value indicates that the corresponding field is numeric, and the at least one metadatum further includes: a sub-datatype index value [f\_value, i\_value] indicative of a type of numeric value of the corresponding field, the sub-datatype index value being selected from a group including at least integer and floating-point numeric value types. (See column 24, lines 31-42 "The f\_value is the value-type of the Rule Step output if the value\_type is float. The i\_value is the value type of the Rule Step output, if the value\_type is integer.") It would have been obvious to one with ordinary skill in the art to add a field storing the datatype in order to appropriately use the data. By knowing the datatype, differing datatypes will not be erroneously combined. It is for this reason that one of ordinary skill in the art would

have been motivated to include the datatype index value indicates that the corresponding field is numeric, and the at least one metadatum further includes: a sub-datatype index value indicative of a type of numeric value of the corresponding field, the sub-datatype index value being selected from a group including at least integer and floating-point numeric value types.

25. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 7 above, and further in view of **Ignat et al.** (US 6,611,838).

**Rappold** teaches an abstraction layer for a database substantially as claimed.

**Rappold** fails to teach a category table associating each of the at least two tables with one or more table characteristics. However, **Ignat et al.** teaches a category table [table] associating each of the at least two tables with one or more table characteristics. (See column 6, lines 52-53 and 63-66 "The data access layer contains metadata that describes how to retrieve physical data from data sources...The data access model objects may include among other things, databases, catalogues, schemas, tables, files, columns, data access keys, indexes and data access joins." The table characteristics are interpreted to be found in the data access model based on this description.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine **Rappold** with **Ignat et al.** because they both deal with using metadata based on databases to create extensible functionality. Also, including a category table, which describes the table characteristics, allows the system to be able to determine what functions are available for the table. It is for this reason that one of ordinary skill in

the art would have been motivated to include a category table associating each of the at least two tables with one or more table characteristics.

26. Claims 9, 11, and 12 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 1 above, and further in view of **Harper et al.** (US 5,717,925).

27. Regarding claim 9, **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the at least one metadatum of at least one control record includes a search flag indicative of a type of searching executable on the corresponding field. However, **Harper et al.** teaches the at least one metadatum of at least one control record includes a search flag indicative [fuzzy search flag] of an type of searching executable on the corresponding field. (See column 8, lines 47-48 "...a fuzzy search flag used for the Search function to specify fuzzy or nonfuzzy searching...") It would have been obvious to one with ordinary skill in the art to include a field indicating the search type in the layer as described above in order to facilitate data access and retrieval. It is for this reason that one of ordinary skill in the art would have been motivated to have the at least one metadatum of at least one control record include a search flag indicative of an type of searching executable on the corresponding field.

28. Regarding claim 11, **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the search flag has a value indicating that the corresponding field is searchable by an SQL query, and the at least one metadatum further includes: at least one SQL query format indicator indicative of an allowable SQL query format. However, **Harper et al.** teaches the search flag has a value indicating that the corresponding field is searchable by an SQL query, and the at least one metadatum further includes: at least one SQL query format indicator indicative [table object type] of an allowable SQL query format. (See column 3 lines 49-51 "The database catalog system manages the metadata store as a relational database and provides sort-query-logic (SQL) support for accessing the meta data objects therein..." and see column 5, lines 22-24 "The purpose of the Table object type is to describe the relevant properties of an SQL relational table or a client/server file.") By including a field indicative of the SQL query format and that SQL queries are allowed, the proper method to query will be used, allowing for efficient searching of the information. It is for this reason that one of ordinary skill in the art would have been motivated to include the search flag with a value indicating that the corresponding field is searchable by an SQL query, and the at least one metadatum further includes: at least one SQL query format indicator indicative of an allowable SQL query format.

29. Regarding claim 12, **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach a case-sensitivity indicator that indicates whether searching on the corresponding field is case-sensitive. However,

**Harper et al.** teaches a case-sensitivity indicator [flag] that indicates whether searching on the corresponding field is case-sensitive. (See column 8, lines 45-47 "...a case sensitivity flag used for the Search function to distinguish between upper and lower case property values...") It would have been obvious to one with ordinary skill in the art to include a case sensitivity flag as it is customary to allow the user to search with either case sensitivity enabled or disabled. By including this field, the correct data can be found regardless of the case, if selected. It is for this reason that one of ordinary skill in the art would have been motivated to include a case-sensitivity indicator that indicates whether searching on the corresponding field is case-sensitive.

30. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** in view of **Harper et al.**, as applied to claim 9 above, and further in view of **McElhiney (5,710,915)**. **Rappold** and **Harper et al.** teach an abstraction layer for a database substantially as claimed. **Rappold** and **Harper et al.** fail to teach the search flag has a value indicating that the corresponding field is searchable by a text search, and the at least one metadatum further includes: a text search field region identifier indicating a portion of the corresponding field that is searchable by the text search. However, **McElhiney** teaches the search flag [202] has a value indicating that the corresponding field is searchable by a text search, and the at least one metadatum further includes: a text search field region identifier [block 202] indicating a portion of the corresponding field that is searchable by the text search. (See column 5, lines 19-26 "Referring now to FIG. 2, after entry, block 202 designates a subset of the fields as non-

searchable. As described above, ...the search table contains the fields which may also be used for the selection criteria or aggregation functions.") One with ordinary skill in the art would have included a flag to indicate if a field is searchable in order to facilitate database operations more effectively. By including this field, time would not be wasted searching on a field that is not formatted properly to be searched. It is for this reason that one of ordinary skill in the art would have been motivated to include the search flag having a value indicating that the corresponding field is searchable by a text search, and the at least one metadatum further includes: a text search field region identifier indicating a portion of the corresponding field that is searchable by the text search.

31. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 1 above, and further in view of **Miura et al.** (US 6,629,091). **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the at least one metadatum of at least one control record includes a sort flag identifying whether sorting can be done on the corresponding field. **Miura et al.**, however, teaches the at least one metadatum [field value] of at least one control record includes a sort flag [sort item flag] identifying whether sorting can be done on the corresponding field. (See column 7, lines 16-19 "...the sort item flag directs sorting or not of the field thereof according to the field value in the case of displaying the search result, and has the following meanings. 1: sorting -1: not sorting.") It would have been obvious to one with ordinary skill in the art to include a field to identify whether a field is able to be sorted in order to ensure data integrity. If there was no sort

allowed on the field, an attempted sort could corrupt the data. It is for this reason that one of ordinary skill in the art would have been motivated to have the at least one metadatum of at least one control record include a sort flag identifying whether sorting can be done on the corresponding field.

32. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 1 above, and further in view of **Hayashi et al.** (US 6,014,677). **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the at least one metadatum [flag] of at least one control record [field information] includes a display flag identifying whether the corresponding field is displayable. **Hayashi et al.**, however, teaches the at least one metadatum of at least one control record includes a display flag identifying whether the corresponding field is displayable. (See page 12, line 46-49 "The field information is constituted by an attribute name, an attribute value default, a display flag, an input flag or the line... The display flag assigns whether the field is displayed or not when the tag display is performed.") It would have been obvious to one with ordinary skill in the art to include a display field flag with the abstraction layer because of the usefulness of being able to know if a field is in a displayable format or not. This keeps the output consistent and readable. It is for this reason that one of ordinary skill in the art would have been motivated to have the at least one metadatum of at least one control record includes a display flag identifying whether the corresponding field is displayable.

33. Claim 15 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** applied to claim 1 above, and further in view of **Williams et al.** (US 6,934,696). **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the one or more metadata tables further includes: a displayable table associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual. **Williams et al.**, however, teaches the one or more metadata tables further includes: a displayable table [string table] associating a plurality of display names [text value] with a field of the database through the control key [text\_id] of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual [internationalization]. (See column 27, lines 45-47 "The text\_id is a number used to look up an actual text value in a string table in a separate language module. Text\_ids are used to make internationalization easier.") One with ordinary skill in the art would have recognized the benefit of having different versions of output based on languages for the ease in internationalization of the software. It is for this reason that one of ordinary skill in the art would have been motivated to have the one or more metadata tables further include: a displayable table associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual.

34. Claim 16 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 1 above, and further in view of **Shah et al.** (US 6,041,325). **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach a syntax table associating syntactically valid inputs with a field of the database through the control key of the control record corresponding to the field. **Shah et al.**, however, teaches a syntax table [database] associating syntactically valid inputs with a field of the database through the control key of the control record corresponding to the field. (See column 12, lines 55-59 and 62-63 "Once a service profile is defined, it can be uniquely identified and stored in database and accessed with display manager. Customization available to a service operator or service provider for specific services include the ability to...further restrict valid input criteria...") One of ordinary skill in the art would have recognized the advantage of checking the inputs for validity in order to keep the data consistent and in the proper format. It is for this reason that one of ordinary skill in the art would have been motivated to include a syntax table associating syntactically valid inputs with a field of the database through the control key of the control record corresponding to the field.

35. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** in view of **Shah et al.** (US 6,041,325) as applied to claim 16 above and further in view of **Subramaniam et al.** (US 6,782,383). **Rappold** and **Shah et al.** teach an abstraction layer for a database substantially as claimed. **Rappold** and **Shah et al.** fail to teach a picklist flag [picklist] indicating whether the entries of the syntax table are

displayable as selections of an input of a GUI dialog box. **Subramaniam et al.**, however, teaches a picklist flag indicating whether the entries of the syntax table are displayable as selections of an input of a GUI dialog box. (See column 21 lines 62-67 "In one embodiment, Search Term Separator control includes a picklist with the following options: "All words (AND match)", "Any word (OR match)," and "Exact Phrase". "All words (AND match)" option specifies a search to find records or documents containing every search keyword entered by the user.") Listing which selections are displayable allows the GUI to display only the proper options to the user, avoiding any errors related to the display. It is for this reason that one of ordinary skill in the art would have been motivated to include a picklist flag indicating whether the entries of the syntax table are displayable as selections of an input of a GUI dialog box.

36. Claims 18 and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 1 above, and further in view of **Colbath et al.** (US Patent Application Publication 2004/0199495).

37. Regarding claim 18, **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach an aliases table associating alias names with fields of the database through the control key of the control record corresponding to the field. **Colbath et al.**, however, teaches an aliases table [alias table] associating alias names with fields of the database through the control key [named entity key] of the control record corresponding to the field. (See page 3,

Art Unit: 2167

paragraph [0044] "Alias table may include one record per proper name, where each record may include a named entity key and one or more alias fields.") By including an alias field with the record, one with ordinary skill in the art would recognize the advantage of allowing multiple variations of words or names to be used to refer to the same field. By doing this, the database becomes more user-friendly and adaptable. It is for this reason that one of ordinary skill in the art would have been motivated to include an aliases table associating alias names with fields of the database through the control key of the control record corresponding to the field.

38. Regarding claim 19, **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the aliases table associates a plurality of alias names with at least one field of the database, each of the plurality of alias names having a language parameter associated therewith. **Colbath et al.**, however, teaches the aliases table [translingual table] associates a plurality of alias names with at least one field of the database, each of the plurality of alias names [coreference] having a language parameter associated therewith [named entity key]. (See page 3, paragraph [0045] "Translingual table may include one record per proper name and/or coreference, where each record may include a named entity key and one or more translation fields.") By including the language parameter, one with ordinary skill in the art would recognize the advantage of being able to present the correct version of the word or name for the language chosen. It is for this reason that one of ordinary skill in the art would have been motivated to have the aliases table associate a plurality of alias names with at

Art Unit: 2167

least one field of the database, each of the plurality of alias names having a language parameter associated therewith.

39. Claim 20 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 1 above, and further in view of **Dysart et al.** (US Patent Application Publication 2004/0139070). **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach a patterns table associating one or more search patterns with a field of the database through the control key of the control record corresponding to the field. **Dysart et al.**, however, teaches a patterns table [second execution table] associating one or more search patterns with a field of the database [data input to the parsing table] through the control key of the control record [reference list field] corresponding to the field. (See page 11, paragraph [0142] "Upon execution of a request, the second execution table will use the search patterns to attempt to match the data input to the parsing table(s), so that a match will identify the reference list field to which the data input corresponds.") It would have been obvious to one with ordinary skill in the art that by adding a table for search patterns, the abstract layer would be more advanced and could intelligently handle the input and properly correlate the input with a field. It is for this reason that one of ordinary skill in the art would have been motivated to include a patterns table associating one or more search patterns with a field of the database through the control key of the control record corresponding to the field.

40. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 21 above, and further in view of **Williams et al.** (US 6,934,696). **Rappold** teaches an abstraction layer for a database substantially as claimed. **Rappold** fails to teach the abstraction layer includes at least one translation table that includes equivalent text in a plurality of languages associated with at least one database field, the formulating of the database access command including: accessing the abstraction layer using a key that includes at least a field identifier and a language selection to retrieve the equivalent text in the selected language. **Williams et al.**, however, teaches the abstraction layer includes at least one translation table (string table) that includes equivalent text in a plurality of languages associated with at least one database field (text\_id), the formulating of the database access command including: accessing the abstraction layer using a key (text\_id) that includes at least a field identifier and a language selection to retrieve the equivalent text in the selected language. (See column 27, lines 45-47 "The text\_id is a number used to look up an actual text value in a string table in a separate language module. Text\_ids are used to make internationalization easier.") One with ordinary skill in the art would have recognized the benefit of having different versions of output based on languages for the ease in internationalization of the software. Also, using the "text\_id" as a key to retrieve the equivalent text in the selected language allows for easy modification of the database for additional languages. It is for this reason that one of ordinary skill in the art would have been motivated to have the abstraction layer include at least one translation table that includes equivalent text in a plurality of languages associated with at least one

database field, the formulating of the database access command including: accessing the abstraction layer using a key that includes at least a field identifier and a language selection to retrieve the equivalent text in the selected language.

41. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** in view of **Williams et al.** (US 6,934,696) as applied to claim 22 above, and further in view of **Subramaniam et al.** (US 6,782,383). **Rappold** and **Williams et al.** teach an abstraction layer for a database substantially as claimed. **Rappold** and **Williams et al.** fail to teach a database access operator, the equivalent text being a displayable name for the database access operator. However, **Subramaniam et al.** teaches a database access operator [AND/OR/EXACT], the equivalent text being a displayable name for the database access operator. (See column 21 lines 62-67 "In one embodiment, Search Term Separator control includes a picklist with the following options: "All words (AND match)", "Any word (OR match)," and "Exact Phrase". "All words (AND match)" option specifies a search to find records or documents containing every search keyword entered by the user." Here the database operators are "and", "or", and "Exact Phrase".) Including the displayable names allows for uniformity of the information that is transmitted to the user. Here, some of the operators are provided, but this method allows for expansion as other operators are needed, making the system flexible. It is for this reason that one of ordinary skill in the art would have been motivated to include a database access operator, the equivalent text being a displayable name for the database access operator.

42. Claim 29 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** as applied to claim 28 above, and further in view of **Miura et al.** (US 6,629,091). **Rappold** teaches an article of manufacture substantially as claimed. **Rappold** fails to teach the identified constraint on the database access command is selected from a group consisting of: a text string in a selected language that is incorporated into the database access command, a datatype constraint, a search pattern, a search constraint, a sorting constraint, and a display constraint. **Miura et al.**, however, teaches the identified constraint [flag] on the database access command is selected from a group consisting of: a text string in a selected language that is incorporated into the database access command, a datatype constraint, a search pattern, a search constraint, a sorting constraint [sort item flag], and a display constraint. (See column 7, lines 16-19 "...the sort item flag directs sorting or not of the field thereof according to the field value in the case of displaying the search result, and has the following meanings. 1: sorting -1: not sorting.") It would have been obvious to one with ordinary skill in the art to include a field to identify whether a field is able to be sorted in order to ensure data integrity. If there was no sort allowed on the field, an attempted sort could corrupt the data. It is for this reason that one of ordinary skill in the art would have been motivated to have the identified constraint on the database access command is selected from a group consisting of: a text string in a selected language that is incorporated into the database access command, a datatype constraint, a search pattern, a search constraint, a sorting constraint, and a display constraint.

43. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** applied to claim 31 above, and further in view of **Subramaniam et al.** (US 6,782,383), in view of **Helgeson et al.** (US 6,643,652), in view of **Williams et al.** (US 6,934,696) and in view of **Donohoe et al.** (US 2002/0174196). **Rappold** teaches an abstraction layer substantially as claimed.

**Rappold** fails to teach a modifiable datatypes table associating a plurality of datatype indices with datatypes; a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator; a modifiable displayable table associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual; and a modifiable exchange rate table providing for conversion of monetary units for multilingual users.

However, **Helgeson, et al.** teaches a modifiable datatypes table associating a plurality of datatype indices with datatypes. (See column 16, lines 23-25 “Attr\_type Int Y The number corresponds to the data type of the attribute...” and column 17, lines 15-35 – see generally the chart where different datatypes are defined.) It would have been obvious to one with ordinary skill in the art to use an index value for the datatypes in order to allow new datatypes to be added at any time by just adding another field in the table. It is for this reason that one of ordinary skill in the art would have been motivated

to include a modifiable datatypes table associating a plurality of datatype indices with datatypes.

Also **Subramaniam et al.** teaches a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator. (See column 21 lines 62-67 "In one embodiment, Search Term Separator control includes a picklist with the following options: "All words (AND match)", "Any word (OR match)," and "Exact Phrase". "All words (AND match)" option specifies a search to find records or documents containing every search keyword entered by the user." Here the database operators are "and", "or", and "Exact Phrase".) It would have been obvious to one with ordinary skill in the art that by including a table with the database operations, there could be different output based on the particular data type. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator.

Also, **Williams et al.** teaches a modifiable displayable table [string table] associating a plurality of display names with a field [text\_id] of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual. (See column 27, lines 45-47 "The text\_id is a number used to look up an actual text value in a string table in a separate language module. Text\_ids are used to make internationalization easier.") One with ordinary skill in the art would have recognized the benefit of having different versions of output based on languages for the ease in internationalization of the

software. It is for this reason that one of ordinary skill in the art would have been motivated to have the one or more metadata tables further include: a displayable table associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual.

Finally, **Donahoe et al.** teaches a modifiable exchange rate table [formatted records] providing for conversion of monetary units for multilingual users. (See page 4, paragraph [0034] "In one embodiment, locale-specific 310 information and currency-specific information 320 are formatted records that may be easily imputed and changed by a Web administrator such that no programmer assistance is necessary when adding/deleting locales to/from the present system. And See page 4, paragraph [0035] "For currencies, a similar principle applies, where each currency code may be associated with appropriate settings for precision..., a currency symbol, and whether the currency symbol should appear to the left or the right of the numerical value. It would have been obvious to one with ordinary skill in the art at the time of the invention to combine **Donahoe et al.** with the teachings of **Rappold** because they both deal with using metadata to provide extensibility of a database and by including an exchange rate table for monetary unit storage, multiple locales and languages can be used, providing greater extensibility in the database. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable exchange rate table providing for conversion of monetary units for multilingual users.

44. Claim 33 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** applied to claim 21 above, and further in view of **Subramaniam et al.** (US 6,782,383), in view of **Helgeson et al.** (US 6,643,652), in view of **Williams et al.** (US 6,934,696) and in view of **Donohoe et al.** (US 2002/0174196).

**Rappold** teaches providing one or more modifiable metadata tables containing metadata relating to the database (See FIG. 1 showing the metadata tables), the one or more modifiable metadata tables including at least one of:

a modifiable controls table [10] containing control records corresponding to fields [100] of the database (See page 2, paragraph [0028] "Column 110 holds numbers identifying the individual row 100, similar to a typical relational database." Each of these individual rows correspond to fields of the database.); the control record for each field including at least a control key [110] associating the control record [100] with the field and at least one metadatum [114] corresponding to the field (See page 2, paragraph [0028]. Column 110 holds the control key associating the record with the field. Also, "Column 114 holds attribute IDs representing the attribute type stored on that row." This represents the metadatum corresponding to the field. Column 114 points to the section of the metadata table which holds the attribute names and attribute types.);

wherein the modifiable controls table can be modified to provide different or additional metadata for selectively extending the functionality of said database interface (See page 1, paragraph [0008] "Another technical advantage of particular embodiments of the present invention is that the extensible database system and method enables a user to store new data in a table without having to create new columns or define a new

table using DDL. As a result, the behavior of database applications can be adjusted during run-time, instead of having to restrict the persistence layer processing only to what can be handled by the existing schema.” This shows the modifiable table is extending the functionality of the database.)

**Rappold** fails to teach a modifiable datatypes table associating a plurality of datatype indices with datatypes; a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator; a modifiable displayable table associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual; and a modifiable exchange rate table providing for conversion of monetary units for multilingual users.

However, **Helgeson, et al.** teaches a modifiable datatypes table associating a plurality of datatype indices with datatypes. (See column 16, lines 23-25 “Attr\_type Int Y The number corresponds to the data type of the attribute...” and column 17, lines 15-35 – see generally the chart where different datatypes are defined.) It would have been obvious to one with ordinary skill in the art to use an index value for the datatypes in order to allow new datatypes to be added at any time by just adding another field in the table. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable datatypes table associating a plurality of datatype indices with datatypes.

Also **Subramaniam et al.** teaches a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator. (See column 21 lines 62-67 "In one embodiment, Search Term Separator control includes a picklist with the following options: "All words (AND match)", "Any word (OR match)," and "Exact Phrase". "All words (AND match)" option specifies a search to find records or documents containing every search keyword entered by the user." Here the database operators are "and", "or", and "Exact Phrase".) It would have been obvious to one with ordinary skill in the art that by including a table with the database operations, there could be different output based on the particular data type. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator.

Also, **Williams et al.** teaches a modifiable displayable table [string table] associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual. (See column 27, lines 45-47 "The text\_id is a number used to look up an actual text value in a string table in a separate language module. Text\_ids are used to make internationalization easier.") One with ordinary skill in the art would have recognized the benefit of having different versions of output based on languages for the ease in internationalization of the software. It is for this reason that one of ordinary skill in the art would have been motivated to have the one or more metadata tables further include: a displayable table associating a plurality

Art Unit: 2167

of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual.

Finally, **Donahoe et al.** teaches a modifiable exchange rate table [formatted records] providing for conversion of monetary units for multilingual users. (See page 4, paragraph [0034] "In one embodiment, locale-specific 310 information and currency-specific information 320 are formatted records that may be easily imputed and changed by a Web administrator such that no programmer assistance is necessary when adding/deleting locales to/from the present system. And See page 4, paragraph [0035] "For currencies, a similar principle applies, where each currency code may be associated with appropriate settings for precision..., a currency symbol, and whether the currency symbol should appear to the left or the right of the numerical value. It would have been obvious to one with ordinary skill in the art at the time of the invention to combine **Donahoe et al.** with the teachings of **Rappold** because they both deal with using metadata to provide extensibility of a database and by including an exchange rate table for monetary unit storage, multiple locales and languages can be used, providing greater extensibility in the database. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable exchange rate table providing for conversion of monetary units for multilingual users.

45. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Rappold** applied to claim 27 above, and further in view of **Subramaniam et al.** (US

6,782,383), in view of **Helgeson et al.** (US 6,643,652), in view of **Williams et al.** (US 6,934,696) and in view of **Donohoe et al.** (US 2002/0174196).

**Rappold** teaches one or more modifiable metadata tables containing metadata relating to the database (See FIG. 1 showing the metadata tables), the one or more modifiable metadata tables including at least one of:

a modifiable controls table [10] containing control records [100] corresponding to fields of the database (See page 2, paragraph [0028] "Column 110 holds numbers identifying the individual row 100, similar to a typical relational database." Each of these individual rows correspond to fields of the database.); the control record for each field including at least a control key [110] associating the control record [100] with the field and at least one metadatum [114] corresponding to the field (See page 2, paragraph [0028]. Column 110 holds the control key associating the record with the field. Also, "Column 114 holds attribute IDs representing the attribute type stored on that row." This represents the metadatum corresponding to the field. Column 114 points to the section of the metadata table which holds the attribute names and attribute types.);

wherein the modifiable controls table can be modified to provide different or additional metadata for selectively extending the functionality of said database interface (See page 1, paragraph [0008] "Another technical advantage of particular embodiments of the present invention is that the extensible database system and method enables a user to store new data in a table without having to create new columns or define a new table using DDL. As a result, the behavior of database applications can be adjusted during run-time, instead of having to restrict the persistence layer processing only to

what can be handled by the existing schema.” This shows the modifiable table is extending the functionality of the database.)

**Rappold** fails to teach a modifiable datatypes table associating a plurality of datatype indices with datatypes; a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator; a modifiable displayable table associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual; and a modifiable exchange rate table providing for conversion of monetary units for multilingual users.

However, **Helgeson, et al.** teaches a modifiable datatypes table [chart] associating a plurality of datatype indices with datatypes. (See column 16, lines 23-25 “Attr\_type Int Y The number corresponds to the data type of the attribute...” and column 17, lines 15-35 – see generally the chart where different datatypes are defined.) It would have been obvious to one with ordinary skill in the art to use an index value for the datatypes in order to allow new datatypes to be added at any time by just adding another field in the table. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable datatypes table associating a plurality of datatype indices with datatypes.

Also **Subramaniam et al.** teaches a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator. (See column 21 lines 62-67 “In one embodiment, Search Term Separator

control includes a picklist with the following options: "All words (AND match)", "Any word (OR match)," and "Exact Phrase". "All words (AND match)" option specifies a search to find records or documents containing every search keyword entered by the user." Here the database operators are "and", "or", and "Exact Phrase".) It would have been obvious to one with ordinary skill in the art that by including a table with the database operations, there could be different output based on the particular data type. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable operators table associating a database operation with a database type index value and with a corresponding display operator.

Also, **Williams et al.** teaches a modifiable displayable table [string table] associating a plurality of display names [actual text] with a field of the database through the control key [text\_id] of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual. (See column 27, lines 45-47 "The text\_id is a number used to look up an actual text value in a string table in a separate language module. Text\_ids are used to make internationalization easier.") One with ordinary skill in the art would have recognized the benefit of having different versions of output based on languages for the ease in internationalization of the software. It is for this reason that one of ordinary skill in the art would have been motivated to have the one or more metadata tables further include: a displayable table associating a plurality of display names with a field of the database through the control key of the controls table, the plurality of display names each corresponding to a different language whereby the display name is multilingual.

Finally, **Donahoe et al.** teaches a modifiable exchange rate table [formatted records] providing for conversion of monetary units for multilingual users. (See page 4, paragraph [0034] "In one embodiment, locale-specific 310 information and currency-specific information 320 are formatted records that may be easily imputed and changed by a Web administrator such that no programmer assistance is necessary when adding/deleting locales to/from the present system. And See page 4, paragraph [0035] "For currencies, a similar principle applies, where each currency code may be associated with appropriate settings for precision..., a currency symbol, and whether the currency symbol should appear to the left or the right of the numerical value. It would have been obvious to one with ordinary skill in the art at the time of the invention to combine **Donahoe et al.** with the teachings of **Rappold** because they both deal with using metadata to provide extensibility of a database and by including an exchange rate table for monetary unit storage, multiple locales and languages can be used, providing greater extensibility in the database. It is for this reason that one of ordinary skill in the art would have been motivated to include a modifiable exchange rate table providing for conversion of monetary units for multilingual users.

### ***Conclusion***

46. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.


**Gorelik et al.** teaches an abstraction layer independent from the data source.


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis L. Vautrot whose telephone number is 571-272-2184. The examiner can normally be reached on Monday-Friday 8:30-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dv  
14 August 2006

  
JOHN COTTINGHAM  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100

 18 August 2006